LOOK BEFORE YOU LOCALIZE: ACTIVE TARGET LOCALIZATION FOR AERIAL VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In language-goal aerial navigation, a UAV progressively acquires observations to localize a target described in a language instruction by its visual attributes and relations to surrounding landmarks. A target may already be visible while the relational, ordinal, or landmark context needed to distinguish it from plausible alternatives remains outside the observed region. This creates an important decision during navigation: whether to localize from the current visual context or continue observing the environment. We propose a closed-loop active target localization framework that makes this decision explicit through two complementary agents. The Selective Target Localization Agent either predicts a target location or defers localization based on the accumulated visual context, while providing a rationale associated with the deferral. The Rationale-Guided Exploration Agent conditions on this rationale and the shared spatial context to select the next exploratory movement. New observations update a world-aligned spatial memory, allowing localization to be reassessed using the retained visual context. Experiments on CityNav show consistent improvements across seen and unseen splits. On the original Test Unseen split, our method achieves 42.57% SR and 30.95% SPL, exceeding the best compared automated SR and SPL by 16.67% and 11.32%, respectively.

1 INTRODUCTION

Vision-and-language navigation (VLN) enables embodied agents to navigate an environment by jointly reasoning over natural-language instructions and visual observations (Anderson et al., 2018). We consider language-goal aerial navigation, in which an unmanned aerial vehicle (UAV) navigates to a destination specified by a natural-language description rather than a prescribed route (Lee et al., 2025). Since the target coordinates are not provided, the UAV grounds the description in the environment to identify a navigation goal, making target localization an integral part of the task.

Unlike visual grounding with a fixed input, target localization during navigation relies on observations acquired progressively through movement. A target may already be visible while the contextual cues needed to distinguish it from similar alternatives remain unobserved. As illustrated in Figure 1, locating "the first red car from the right in the third row" depends on the row layout, vehicle ordering, and the right boundary of the row, which may not all be captured in a partial view of the parking lot. Without this context, committing to a plausible candidate prematurely can send the UAV to an incorrect location. A central challenge is therefore *deciding whether to localize from the visual context accumulated so far or to acquire further observations before making a prediction.*

Recent aerial VLN methods employ pretrained language and vision-language models for semantic grounding, spatial reasoning, and action prediction (Cai et al., 2025; Xu et al., 2026; Tong et al., 2026). Hierarchical planning and historical memory support long-range navigation (Zhang et al., 2025b), while target verification and progress monitoring help assess an agent's trajectory (Zheng et al., 2026a). Despite these advances, a visible candidate may remain ambiguous when instruction-relevant landmarks, spatial relations, or ordinal boundaries lie outside the observed region.

We address this gap through *active target localization*, a two-agent framework that explicitly alternates between target localization and further observation. The *Selective Target Localization Agent* either predicts a target location or defers with a rationale based on the accumulated visual con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

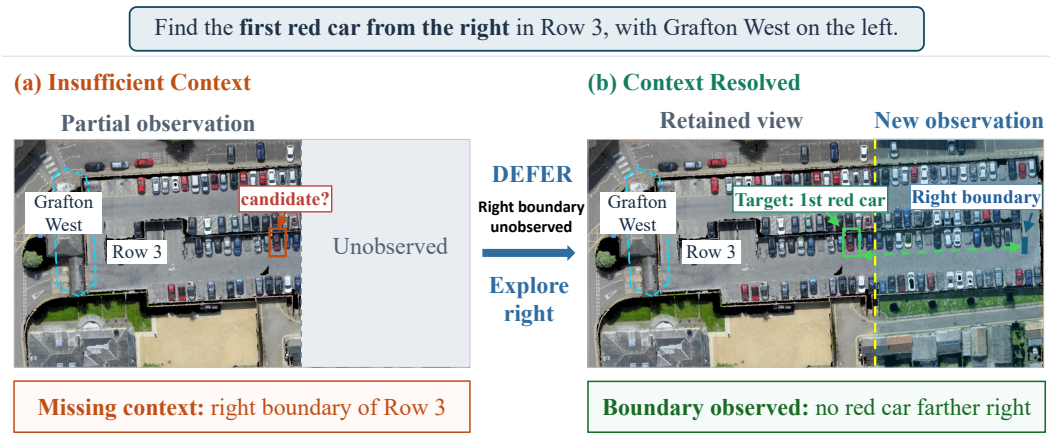

Figure 1: **A target can be visible before it is identifiable.** A partial observation contains a plausible target but omits the right boundary of Row 3, leaving the ordinal cue unresolved. After observing the right boundary, the agent can check whether another red car lies farther to the right and identify the referred target in this example.

text. The *Rationale-Guided Exploration Agent* uses this rationale and the current spatial context to select the next search movement. Both agents operate on a shared world-aligned spatial memory that preserves spatial relationships across observations. After each exploratory movement, the new observation is incorporated into memory and the localization agent reassesses the updated context. This interaction continues until a target location is predicted or the exploration budget is exhausted, forming a closed localization–exploration loop.

To learn the localize or defer behavior, we train the localization agent with counterfactual sparse-view supervision, which varies the observed regions for the same scene–instruction pair. Operational criteria based on target visibility and selected contextual cues provide localization or deferral targets, while condition specific rationales supervise the deferral outputs. At inference, the generated rationale conditions the exploration agent's next search movement.

Our main contributions are:

- We propose a closed-loop dual-agent framework that couples selective target localization with rationale-guided exploration, using deferral rationales to condition the selection of subsequent observations.
- We introduce counterfactual sparse-view supervision that trains the localization agent on partial views with operationally assigned localization or deferral targets and rationales associated with the deferral conditions.
- Our method achieves the best performance among the compared automated methods across the seen and unseen splits of CityNav. On the original Test Unseen split, it achieves 42.57% SR and 30.95% SPL, exceeding the strongest compared automated baselines by 16.67% and 11.32%, respectively.

2 Related Work

Aerial Vision-and-Language Navigation. Vision-and-language navigation (VLN) combines language grounding with sequential action selection (Anderson et al., 2018). AerialVLN extends instruction following to UAVs (Liu et al., 2023), and recent surveys, platforms, and benchmarks further broaden the aerial VLN landscape (Yao et al., 2025; Gao et al., 2026; Wang et al., 2025), while CityNav studies city-scale language-goal navigation with geographic landmark priors (Lee et al., 2025). Recent methods improve VLM-based reasoning and action prediction (Cai et al., 2025; Zhang et al., 2025a), enhance history modeling (Ding et al., 2026), and coordinate global planning with fine-grained control (Fan et al., 2026; Zhang et al., 2025b). GeoNav organizes landmark navigation, target search, and localization through geographic priors and hierarchical scene graphs (Xu et al., 2026). OnFly separates target generation from progress monitoring and verifies candidates for instruction consistency and geometric safety (Zheng et al., 2026a), while ViSA uses structured visual

prompting to verify candidates, defer uncertain decisions, and refine detection prompts (Tong et al., 2026). Our work focuses on the decision of whether accumulated views support localization. Counterfactual sparse-view supervision directly trains the localization agent to predict a target or defer as the available context changes for the same scene–instruction pair.

Language-Grounded Target Localization. Visual grounding localizes language-referred objects from appearance and contextual relationships in 2D images (Yu et al., 2018), reconstructed 3D scenes (Chen et al., 2020), and city-scale point clouds (Miyanishi et al., 2023). Recent methods extend grounding through multimodal spatial reasoning: SeeGround selects query-aligned rendered views (Li et al., 2025b); CityAnchor performs coarse-to-fine city-scale localization (Li et al., 2025a); GeoProg3D composes geographic operations over 3D language fields (Yasuki et al., 2025); and CityVG reasons over scene-graph-aligned candidates (Zhang & Wang, 2026). Embodied grounding benchmarks such as REVERIE further require navigation to locate the referred object (Qi et al., 2020). These methods ground targets within supplied images or scene representations. Our framework accumulates visual context across views, permits localization to be deferred, and uses the deferral rationale to guide the acquisition of another observation.

Active Visual Search. Active visual search selects observations to discover and localize targets under partial observability (Andreopoulos & Tsotsos, 2009). L3MVN and VLFM use language or visual-semantic priors to prioritize frontiers (Yu et al., 2023; Yokoyama et al., 2024). Recent work addresses uncertainty and perceptual ambiguity. Bajpai et al. (2025) incorporate semantic uncertainty into mapping and frontier planning, while MAG-Nav uses viewpoint adjustment and memory backtracking to reassess uncertain candidates (Zhang et al., 2025c). 3DGSNav builds an incremental 3D Gaussian representation and switches viewpoints to reverify targets (Zheng et al., 2026b), whereas ProFocus checks information sufficiency through targeted queries of panorama crops (Xue et al., 2026). These works demonstrate the value of active perception and context assessment. We learn selective localization under incomplete instruction-dependent context and condition physical exploration on the deferral rationale, even when a plausible target is visible.

3 METHOD

3.1 PROBLEM FORMULATION AND FRAMEWORK OVERVIEW

Problem formulation. We study language-goal aerial navigation with target-blind geographic landmark priors (Lee et al., 2024). An episode is defined as $\mathcal{E} = (\ell, \mathbf{s}_{\text{init}}, \mathcal{G})$, where ℓ is the target instruction, $\mathbf{s}_{\text{init}}$ is the initial UAV pose, and $\mathcal{G} = \{(u_j, P_j)\}_{j=1}^{N}$ contains landmark labels and polygonal contours in world coordinates. The target location is excluded from $\mathcal{G}$ and remains unknown at inference. The instruction-referenced subset $\mathcal{G}(\ell)$ initializes the search and is provided to the exploration agent. At step t, the UAV at pose $\mathbf{s}_t$ acquires a local top-down RGB observation I_t. Given the instruction, landmark priors, and accumulated observations, the system either predicts a target location or defers localization to acquire another observation. The search allows at most T exploratory movements. Once a target location $\hat{\mathbf{p}}_t$ is predicted, the UAV navigates to it and stops.

Framework overview. Our framework implements aerial target search as a closed-loop interaction between two agents operating on shared spatial context. It first initializes the visual search around instruction-referenced geographic landmarks and incrementally integrates newly acquired observations in a world-aligned representation. The Selective Target Localization Agent then emits either a target box or a deferral response. The deferral rationale describes the context associated with the agent's decision. The Rationale-Guided Exploration Agent conditions its next movement on this rationale and the retained spatial context. The resulting observation is incorporated into memory, and the localization decision is repeated until a box is emitted or the exploration budget is exhausted. Figure 2 summarizes the resulting closed loop.

3.2 WORLD-ALIGNED SPATIAL MEMORY

Language-grounded target search can require spatial relations that extend beyond a single local observation. Because top-down views acquired from different UAV positions are expressed in separate image coordinate frames, processing them independently fragments instruction-relevant context across search steps. We therefore maintain a world-aligned spatial memory by using known UAV

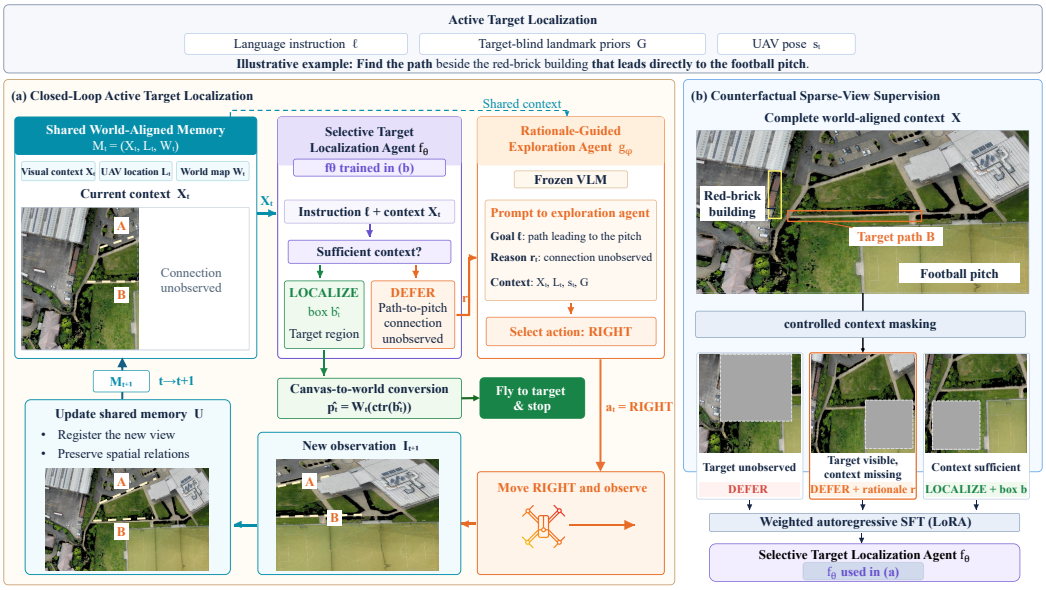

Figure 2: **Overview of the proposed active target localization framework.** Given a language instruction, target-blind landmark priors, and the UAV pose, the two agents operate on a shared world-aligned spatial memory. (a) The Selective Target Localization Agent receives the instruction and accumulated visual context, and either predicts a target box or defers with a rationale. Following a deferral, the frozen Rationale-Guided Exploration Agent uses the rationale, shared context, UAV pose, and landmark priors to select the next search movement. The resulting observation is registered in memory, and localization is reassessed. Once a target box is predicted, its center is converted to a world-coordinate goal for terminal navigation. (b) Counterfactual sparse-view supervision used in the closed loop in (a) constructs partial views by masking regions of a complete world-aligned context while keeping the scene, instruction, and target fixed. A masked view receives `DEFER` supervision when the target is unobserved or selected instruction-relevant context is missing, and `LOCALIZE` supervision when the target and selected contextual cues are retained.

poses and observation to world correspondences to register each new view at its corresponding location on a common canvas. Within the retained canvas window, the memory provides a shared visual representation for the two agents: the localization agent receives the retained canvas, while the exploration agent uses it, together with the deferral response, to select the next search movement.

At search step t, the accumulated spatial memory is represented as

$$\mathcal{M}_t = (X_t, L_t, \mathcal{W}_t), \tag{1}$$

where $X_t \in \mathbb{R}^{H\times W\times 3}$ is a fixed-resolution, world-aligned RGB canvas covering a ground-plane region, and $L_t \in \{0,1\}^{H\times W}$ marks the UAV location. Before registration, each observation is counter-rotated by the UAV heading and placed on the canvas using its pose, aligning views acquired from different positions and headings.

The canvas-to-world correspondence is defined as

$$\mathcal{W}_t : \Omega \to \mathbb{R}^2, \tag{2}$$

where Ω is the pixel domain of X_t. For a canvas pixel $\mathbf{q} = (u, v)$, $\mathcal{W}_t(\mathbf{q})$ returns its ground-plane world coordinate. This mapping is determined by the canvas origin, orientation, and metric resolution, and is used for view registration and target-coordinate conversion.

Given a new observation I_{t+1} at pose $\mathbf{s}_{t+1}$, the memory is updated as

$$\mathcal{M}_{t+1} = \mathcal{U}(\mathcal{M}_t, I_{t+1}, \mathbf{s}_{t+1}), \tag{3}$$

where $\mathcal{U}$ registers the observation and updates L_t and, when the canvas shifts, $\mathcal{W}_t$. The initial memory $\mathcal{M}_0$ is constructed from $(I_0, \mathbf{s}_0)$.

To bound the visual input, the fixed-size canvas follows the active search area. When it shifts, overlapping observations are retained, content outside the new window is discarded, and unobserved regions are filled with a fixed placeholder.

3.3 Selective Target Localization Agent

The localization agent augments target prediction with a decision to predict or defer under partial observation. A visible candidate may remain ambiguous when a referenced landmark, spatial relation, or ordinal boundary lies outside the observed region. We train the agent on constructed sparse views to predict a target box when the target and selected instruction-relevant cues are visible, and otherwise to defer with a rationale describing the missing target region or context.

We instantiate the Selective Target Localization Agent as a vision-language model f_θ, initialized from a pretrained VLM and fine-tuned to generate a structured localization response. Given the instruction ℓ and accumulated visual context X_t, the model autoregressively produces an output sequence y_t of one of two forms:

$$y_t = f_\theta(\ell, X_t), \qquad y_t \in \big\{[\texttt{LOCALIZE}; \hat{\mathbf{b}}_t], [\texttt{DEFER}; r_t]\big\}. \tag{4}$$

where $[\cdot;\cdot]$ denotes the response schema. A `LOCALIZE` response contains a predicted target bounding box $\hat{\mathbf{b}}_t \in [0,1]^4$, expressed in normalized canvas coordinates. The predicted box is deterministically mapped to pixel coordinates before conversion to the world frame. A `DEFER` response contains structured deferral information together with a natural-language rationale describing the context associated with the decision.

Counterfactual sparse-view supervision. To train the localization agent to choose between target prediction and deferral under partial observation, we introduce counterfactual sparse-view supervision (CSS). CSS varies the observed region while keeping the scene, instruction, and target fixed. From a complete canvas X, a spatial mask m produces

$$\widetilde{X} = \mathcal{M}(X; m), \tag{5}$$

where $\mathcal{M}$ retains connected regions and fills the rest with a fixed placeholder. Each masked view receives a training response

$$y = \begin{cases} [\texttt{LOCALIZE}; \mathbf{b}], & \text{if the target and selected contextual cues are observed,} \\ [\texttt{DEFER}; r], & \text{otherwise,} \end{cases} \tag{6}$$

Here, $\mathbf{b}$ is the ground-truth target box, and r is a template rationale describing the unobserved target region or context. The selected contextual cues include annotated reference landmarks and same-type structural context associated with supported relational or ordinal descriptions. These annotation-derived criteria provide proxy supervision for offline training and are not applied as rules during navigation; they do not exhaustively determine whether a target is uniquely identifiable. Details of mask sampling, contextual criteria, rationale templates, and output serialization are provided in Appendix A. We fine-tune on $\mathcal{D} = \{(\ell_i, \widetilde{X}_i, y_i)\}_{i=1}^N$ using weighted autoregressive cross-entropy:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i \sum_{k=1}^{|y_i|} \log p_\theta\big(y_{i,k} \mid y_{i,<k}, \ell_i, \widetilde{X}_i\big), \tag{7}$$

Here p_θ is the token distribution and w_i upweights target-visible deferrals, including cases where a candidate is visible but the specified context condition is unmet.

3.4 Rationale-Guided Exploration Agent

Following a deferral, the Rationale-Guided Exploration Agent conditions the next search movement on the rationale generated by the localization agent. The rationale describes the context associated with the deferral; the instruction, shared spatial context, UAV pose, and target-blind landmark priors provide information for selecting a movement. We implement this agent as a frozen VLM prompted to select from a bounded action space. The resulting observation is incorporated into shared memory before localization is reassessed.

We restrict the exploration agent to four heading-relative search actions,

$$\mathcal{A}^{\text{search}} = \{\text{forward}, \text{backward}, \text{left}, \text{right}\}, \tag{8}$$

each of which moves the UAV by a fixed distance Δd before acquiring the next observation. We use r_t to denote the natural-language reason field in the deferral response. Given r_t, the frozen exploration VLM g_ϕ predicts

$$a_t = g_\phi\big(\ell, r_t, X_t, L_t, \mathbf{s}_t, \mathcal{G}\big) \in \mathcal{A}^{\text{search}}, \tag{9}$$

where the reason field describes the context to be considered, while the remaining inputs ground that context relative to the UAV. In the exploration prompt, X_t is the current world-aligned canvas and L_t marks the UAV location; $\mathbf{s}_t$ defines the heading-relative action frame, and $\mathcal{G}$ provides the labels and contours of target-blind landmark priors. For a rationale concerning a geographic landmark, the prompt provides its geometry and $\mathbf{s}_t$ to help infer a relative direction. For a rationale concerning local structure, such as an incomplete parking area or an unobserved row origin, the agent can use X_t and L_t to select a direction within the retained window.

Closed-loop execution and termination. After executing a_t, the UAV transitions to pose $\mathbf{s}_{t+1}$ and acquires a new observation I_{t+1}, which is incorporated into the world-aligned spatial memory according to Eq. (3). The localization agent is re-invoked with (ℓ, X_{t+1}). If the localization agent produces a `DEFER` response, the new rationale conditions the next exploration step. If it produces a `LOCALIZE` response with target box $\hat{\mathbf{b}}_t$ at step t, the search loop terminates. Let $\text{pix}_t(\hat{\mathbf{b}}_t)$ denote the deterministic conversion from normalized $[0,1]^4$ coordinates to pixel coordinates on X_t. The center of the resulting pixel-space box is converted into a world-coordinate navigation goal through

$$\hat{\mathbf{p}}_t = \mathcal{W}_t\left(\text{ctr}\left(\text{pix}_t(\hat{\mathbf{b}}_t)\right)\right), \tag{10}$$

where $\text{ctr}(\cdot)$ denotes the center of a pixel-space bounding box. The UAV then navigates from its current pose to $\hat{\mathbf{p}}_t$ and stops. The active localization loop continues for at most T exploratory movements; if no target box is predicted within this budget, the UAV terminates at its current position.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate our method on CityNav (Lee et al., 2025), which contains 32,637 natural-language instruction–trajectory pairs for 5,850 target objects across Birmingham and Cambridge. Its observations are derived from orthographic SensatUrban imagery covering 13 blocks in Birmingham and 33 in Cambridge. CityRefer provides semantic landmark identities and geometric contours as target-blind geographic priors. Following the standard partition, we report results on Validation Seen, Validation Unseen, and Test Unseen, with the unseen splits evaluating generalization to urban environments absent from training. We also train and evaluate on the revised CityNav dataset released with HETT (Ding et al., 2026), which corrects landmark annotations and removes non-navigable instruction–trajectory pairs.

Evaluation metrics. Following the standard evaluation protocol of CityNav, we employ four navigation metrics: Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OSR), and Success weighted by Path Length (SPL). NE measures the Euclidean distance between the final UAV position and the ground-truth target, where a lower value indicates more accurate navigation. SR measures the percentage of episodes in which the UAV terminates within 20m of the target. OSR measures the percentage of episodes in which the UAV reaches within 20m of the target at any point along its executed trajectory, regardless of its final stopping position. SPL further accounts for navigation efficiency by weighting successful episodes according to the ratio between the reference path length and the actually executed path length.

4.2 Comparison with Prior Methods

We evaluate the proposed framework on both the original and revised CityNav datasets, comparing against representative aerial language-goal navigation methods, including Seq2Seq+GSM,

Table 1: Overall performance on the CityNav benchmark. NE is reported in meters; SR, OSR, and SPL are percentages. Bold and underline indicate the best and second-best results among automated methods within each dataset protocol, respectively. Human performance is provided as a reference. * denotes results obtained on the revised CityNav dataset.

	Validation Seen				Validation Unseen				Test Unseen			
Models	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑
Human (Lee et al., 2025)	9.1	89.31	96.40	60.17	9.4	88.39	95.54	62.66	9.8	87.86	95.29	57.04
Random (Lee et al., 2024)	222.3	0.00	1.15	0.00	223.0	0.00	0.90	0.00	208.8	0.00	1.44	0.00
Seq2Seq+GSM (Lee et al., 2025)	58.5	8.43	17.31	7.28	78.6	5.13	10.90	4.65	98.1	3.81	13.92	2.79
CMA+GSM (Lee et al., 2025)	68.0	6.25	13.28	5.40	75.9	4.38	9.29	3.90	94.6	4.68	12.01	4.05
MGP+LLaVA (Lee et al., 2024)	59.7	8.69	35.51	8.28	75.1	5.84	22.19	5.56	93.8	6.38	26.04	6.08
AerialVLN+GSM (Lee et al., 2025)	56.6	10.16	22.20	7.89	72.7	6.35	15.24	5.06	85.1	6.72	18.21	5.16
FlightGPT (Cai et al., 2025)	66.1	17.57	30.26	15.78	68.1	14.69	29.33	13.24	76.2	21.20	35.38	19.24
GeoNav (Xu et al., 2026)	58.6	21.80	39.90	12.30	64.1	16.90	35.50	9.90	73.5	25.90	41.60	16.00
HTNav (Fan et al., 2026)	47.2	28.30	47.92	25.07	59.8	15.85	34.43	13.55	68.5	22.23	41.02	19.63
HETT (Ding et al., 2026)	45.2	25.16	48.40	23.01	62.1	17.48	25.09	14.46	72.9	22.97	39.30	17.01
Ours	**36.4**	**41.82**	**54.98**	**31.18**	**43.4**	**36.59**	**50.13**	**26.64**	**35.7**	**42.57**	**60.90**	**30.95**
HETT* (Ding et al., 2026)	37.2	31.09	51.86	25.76	51.3	19.10	34.78	16.70	40.4	28.90	49.56	23.79
Ours*	**32.9**	**46.80**	**57.85**	**34.52**	**39.8**	**40.67**	**52.28**	**29.58**	**30.7**	**51.45**	**66.33**	**38.33**

Table 2: Selective localization under partial observation. All models use the same inference prompt. Macro-F1 measures the binary LOCALIZE/DEFER decision; FLR measures localization on operational deferral cases, and FLR_2 restricts this metric to target-visible deferral cases. Cond. Acc. measures localization within 20 m among committed localization-target cases, while UDR measures unnecessary deferral on those cases. Always-defer is included only as a non-learned reference.

Model	Macro-F1 (%)↑	FLR (%)↓	FLR_2 (%)↓	Cond. Acc. (%)↑	UDR (%)↓
Base VLM (2B)	51.82	20.23	21.21	27.22	76.04
Bbox-only SFT	56.08	37.79	44.56	42.81	**44.59**
Ours	**65.81**	**7.23**	**14.14**	**45.96**	64.47
Always-defer (ref.)	*42.49*	*0.00*	*0.00*	*N/A*	*100.00*

CMA+GSM, and AerialVLN+GSM (Anderson et al., 2018; Liu et al., 2023; Lee et al., 2025); the map-based planner MGP (Lee et al., 2024); and recent VLM-based or geospatial approaches FlightGPT, GeoNav, HTNav, and HETT (Cai et al., 2025; Xu et al., 2026; Fan et al., 2026; Ding et al., 2026). Random is included as a lower-bound reference, while human performance under the CityNav evaluation protocol is reported as a reference. HETT* and Ours* denote results on the revised CityNav dataset.

As shown in Table 1, our method improves both SR and SPL across the evaluated splits. The concurrent improvements in SR and SPL suggest that the higher success rate is not achieved simply by taking longer search trajectories. On the original Test Unseen split, our method achieves 42.57% SR and 30.95% SPL, exceeding the best reported results among the compared automated methods by 16.67% and 11.32%, respectively. The gains on both seen and unseen validation splits further indicate that the benefit is not limited to familiar environments. Under the revised CityNav protocol, Ours* outperforms HETT* by 22.55% in SR and 14.54% in SPL, showing that the improvement remains after correcting landmark annotations and removing non-navigable episodes.

4.3 Selective Localization under Partial Observation

To evaluate the Selective Target Localization Agent under partial observation, we compare three variants using the same inference prompt, considering both LOCALIZE/DEFER decisions and target localization accuracy. **Base VLM** uses the pretrained model without task-specific fine-tuning. **Bbox-only SFT** learns target-box prediction from complete views without LOCALIZE/DEFER supervision. **Ours** uses counterfactual sparse-view supervision (CSS) to learn this decision from partial views. Following the CSS construction, a sparse view receives a LOCALIZE target when the target and selected instruction-relevant cues are visible, and a DEFER target otherwise.

Table 3: Component ablations on the original CityNav benchmark.

	Validation Seen				Validation Unseen				Test Unseen			
Variant	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑	NE↓	SR↑	OSR↑	SPL↑
Full model	**36.37**	**41.82**	**54.98**	31.18	43.42	**36.59**	**50.13**	26.64	**35.74**	**42.57**	**60.90**	30.95
w/o Selective Supervision	41.11	30.04	47.57	25.57	**43.40**	32.80	46.46	**28.33**	39.52	35.11	57.22	30.28
w/o Rationale Guidance	37.35	38.99	53.16	29.12	44.10	34.77	49.05	25.77	40.46	37.09	59.10	27.95
w/o Spatial Memory	51.99	39.15	52.47	**33.42**	62.47	32.36	46.79	27.45	53.28	38.14	58.08	**32.99**

Table 4: Controlled analysis of rationale content on the original CityNav Test Unseen split. "Reveal" denotes the rate at which the next observation reveals the context associated with the reference deferral condition under the evaluation protocol; "One-step" is the one-step DEFER-to-LOCALIZE rate; "Expl." is the mean number of executed exploratory movements per episode.

Rationale variant	Reveal↑ (%)	One-step↑ (%)	Expl.↓	SR↑ (%)
Original rationale (Ours)	**33.25**	**42.03**	**1.22**	**42.57**
No rationale	14.89	29.83	1.52	37.09
Generic rationale	15.01	29.61	1.52	37.10
Shuffled rationale	10.40	25.58	2.22	34.36

Macro-F1 is the primary metric for this binary decision. FLR measures LOCALIZE predictions on DEFER views, while FLR_2 considers only cases where the target is visible but a selected instruction-relevant cue is missing. UDR measures DEFER predictions on LOCALIZE views, and conditional localization accuracy evaluates target prediction when the agent selects LOCALIZE. Appendix A provides human-annotation results, metric definitions, and evaluation details.

As shown in Table 2, CSS improves Macro-F1 by 9.73% over bbox-only SFT. Both FLR and FLR_2 decrease by about 30%, showing that the agent predicts targets less often on views labeled DEFER, including cases where the target is visible but a selected instruction-relevant cue is missing. Conditional localization accuracy improves slightly, indicating that this reduction does not lower target prediction accuracy when the agent selects LOCALIZE. However, the higher UDR shows that the agent becomes more conservative and sometimes defers on LOCALIZE views. The always-defer reference confirms that FLR cannot be considered alone, as it misses every LOCALIZE case and achieves a lower Macro-F1. The human-annotation results in Appendix A.6 show the same pattern.

4.4 Component Ablations

We examine the effects of three components on navigation in Table 3: selective supervision for localization or deferral, rationale conditioning for exploration, and spatial memory for retaining context across views. "*w/o Selective Supervision*" replaces CSS with complete view, bounding box only fine-tuning on the same source episodes, while retaining the inference interface that permits both LOCALIZE and DEFER. "*w/o Rationale Guidance*" removes rationale r_t from the exploration agent input while retaining the CSS trained localization agent and all other exploration inputs. "*w/o Spatial Memory*" provides both agents with only the current observation, while retaining CSS and rationale guided exploration.

Effect of selective supervision. Replacing CSS with complete view, box only training reduces SR across all three splits. On Test Unseen, SR decreases from 42.57% to 35.11%, a drop of 7.46%, which indicates that CSS better prepares the localization agent for the partial observations encountered during closed-loop search.

Effect of rationale guidance. Removing the rationale from the exploration-agent input reduces Test Unseen SR by 5.48% and SPL by 3.00% in Table 3, showing that the rationale contributes to end-to-end navigation. The controlled analysis in Table 4 further explains this improvement: the original rationale more than doubles the context-reveal rate and achieves the highest one-step DEFER-to-LOCALIZE rate with the fewest exploratory movements. These results indicate that the rationale helps the exploration agent select observations that support the next localization decision.

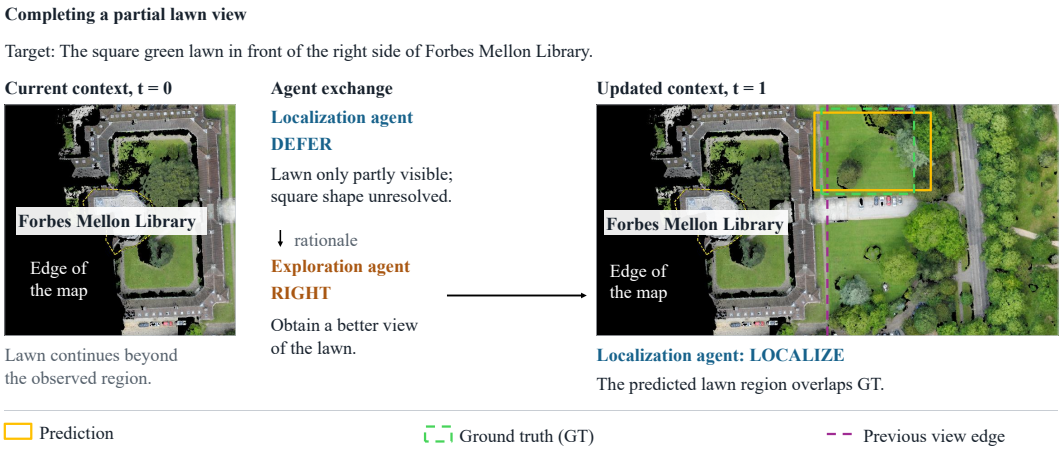

Figure 3: **Qualitative example of closed-loop target localization on CityNav.** At $t = 0$, the localization agent defers because only part of the lawn is visible, and its rationale guides the exploration agent to move `RIGHT`. The resulting observation is registered in the shared spatial memory, after which the localization agent predicts the lawn region at $t = 1$. Yellow solid and green dashed boxes denote the prediction and ground truth, respectively; magenta dashed lines mark the boundary of the previous view. Search actions are relative to the UAV heading.

A generic rationale performs similarly to no rationale, while shuffled rationales perform worse, showing that the gain depends on rationale content specific to the current episode.

Effect of spatial memory. Spatial memory improves SR and reduces NE across all three splits in Table 3. On Test Unseen, removing memory reduces SR from 42.57% to 38.14% and increases NE from 35.74 m to 53.28 m. However, the memory-free variant achieves higher SPL across all three splits, including 32.99% versus 30.95% on Test Unseen. One possible explanation is that retained context allows the agent to localize targets in previously observed regions, potentially requiring return travel during terminal navigation and increasing the executed path length.

4.5 Qualitative Results

Figure 3 illustrates how the two agents resolve a partially observed target. At $t = 0$, the instruction refers to the square lawn beside Forbes Mellon Library, but the lawn extends beyond the current view, leaving its shape and extent unclear. The localization agent therefore defers and identifies the incomplete lawn view in its rationale. The exploration agent selects `RIGHT`, revealing the previously unobserved part of the lawn. After the new observation is registered in the shared spatial memory, the library remains available as a reference landmark while a more complete lawn region becomes visible. The localization agent then predicts a region that overlaps the ground truth. This example shows how the rationale directs exploration toward the unresolved part of the scene and how spatial memory supports localization across observations. Our method sometimes still fails after exploration, typically when visually similar candidates remain or when spatial and ordinal relations are incorrectly interpreted. More successful and failure cases are provided in Appendix C.

5 Conclusion

In this work, we study language-goal aerial navigation as a sequence of localization-or-deferral decisions under partial observation. A visible candidate may remain ambiguous when instruction-relevant landmarks, spatial relations, or ordinal boundaries lie outside the observed region. To address this problem, we introduce a closed-loop framework comprising a Selective Target Localization Agent and a Rationale-Guided Exploration Agent. Counterfactual sparse-view supervision trains the localization agent to predict a target or defer as the observed context changes for the same scene–instruction pair. When localization is deferred, the generated rationale guides the next exploratory movement. New observations are integrated into a shared world-aligned spatial memory, allowing the target decision to be reconsidered as the search progresses. Experiments on the original and revised CityNav protocols show consistent improvements across seen and unseen splits.

AI USE STATEMENT

We used generative AI tools to polish the language and improve the readability of the manuscript. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

This work uses existing aerial navigation datasets, with additional manual annotations for evaluation. Real-world deployment should address privacy and operational safety concerns and comply with applicable regulations.

REPRODUCIBILITY STATEMENT

Section 3 describes the proposed framework, and Section 4.1 specifies the datasets and evaluation metrics. Appendix A details the training data construction and selective localization evaluation protocols, while Appendix E provides the training and inference settings.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Alexander Andreopoulos and John K. Tsotsos. A theory of active object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

Utkarsh Bajpai, Julius Rückin, Cyrill Stachniss, and Marija Popović. Uncertainty-informed active perception for open vocabulary object goal navigation. In *Proceedings of the European Conference on Mobile Robots*, 2025.

Hengxing Cai, Jinhan Dong, Jingjun Tan, Jingcheng Deng, Sihang Li, Zhifeng Gao, Haidong Wang, Zicheng Su, Agachai Sumalee, and Renxin Zhong. FlightGPT: Towards generalizable and interpretable UAV vision-and-language navigation with vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. In *Computer Vision – ECCV 2020*, 2020.

Xichen Ding, Jianzhe Gao, Cong Pan, Wenguan Wang, and Jie Qin. History-enhanced two-stage transformer for aerial vision-and-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Chengjie Fan, Cong Pan, Zijian Liu, Ningzhong Liu, and Jie Qin. HTNav: A hybrid navigation framework with tiered structure for urban aerial vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, Yiwen Tang, Yuhang Tang, Shuai Liang, Songyi Zhu, Ziqin Xiong, Yifei Su, Xinyi Ye, Jianan Li, Yan Ding, Dong Wang, Xuelong Li, Zhigang Wang, and Bin Zhao. OpenFly: A comprehensive platform for aerial vision-language navigation. In *International Conference on Learning Representations*, 2026.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. CityNav: Language-goal aerial navigation dataset with geographic information. *arXiv preprint arXiv:2406.14240*, 2024.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. CityNav: A large-scale dataset for real-world aerial navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Jinpeng Li, Haiping Wang, Jiabin Chen, Yuan Liu, Zhiyang Dou, Yuexin Ma, Sibei Yang, Yuan Li, Wenping Wang, Zhen Dong, and Bisheng Yang. CityAnchor: City-scale 3D visual grounding with multi-modality LLMs. In *International Conference on Learning Representations*, 2025a.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. SeeGround: See and ground for zero-shot open-vocabulary 3D visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. AerialVLN: Vision-and-language navigation for UAVs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Taiki Miyanishi, Fumiya Kitamori, Shuhei Kurita, Jungdae Lee, Motoaki Kawanabe, and Nakamasa Inoue. CityRefer: Geography-aware 3D visual grounding dataset on city-scale point cloud data. In *Advances in Neural Information Processing Systems*, 2023.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. REVERIE: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Haoyu Tong, Xiangyu Dong, Xiaoguang Ma, Haoran Zhao, Yaoming Zhou, and Chenghao Lin. ViSA-enhanced aerial VLN: A visual-spatial reasoning enhanced framework for aerial vision-language navigation. *arXiv preprint arXiv:2603.08007*, 2026.

Xiangyu Wang, Donglin Yang, Yue Liao, Wenhao Zheng, Wenjun Wu, Bin Dai, Hongsheng Li, and Si Liu. UAV-Flow Colosseo: A real-world benchmark for flying-on-a-word UAV imitation learning. In *Advances in Neural Information Processing Systems*, 2025.

Haotian Xu, Yue Hu, Chen Gao, Zhengqiu Zhu, Yong Zhao, and Quanjun Yin. GeoNav: Empowering MLLMs with dual-scale geospatial reasoning for language-goal aerial navigation. *Pattern Recognition*, 2026.

Wei Xue, Mingcheng Li, Xuecheng Wu, Jingqun Tang, Dingkang Yang, and Lihua Zhang. ProFocus: Proactive perception and focused reasoning in vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Fanglong Yao, Youzhi Liu, Wenyi Zhang, Zhengqiu Zhu, Chenglong Li, Nayu Liu, Peng Hu, Yuanchang Yue, Kaiwen Wei, Xin He, Xudong Zhao, Zihan Wei, Haotian Xu, Zhiyuan Wang, Gujie Shao, Liu Yang, Dan Zhao, and Yong Yang. AeroVerse-Review: Comprehensive survey on aerial embodied vision-and-language navigation. *The Innovation Informatics*, 2025.

Shunsuke Yasuki, Taiki Miyanishi, Nakamasa Inoue, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Masato Taki, and Yutaka Matsuo. GeoProg3D: Compositional visual reasoning for city-scale 3D language fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-language frontier maps for zero-shot semantic navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2024.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3MVN: Leveraging large language models for visual target navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. MAttNet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Jianjun Zhang and Hanli Wang. CityVG: Contrastive fine-tuning and reward-based chain-of-thought reasoning for zero-shot city-scale 3D visual grounding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

Lingfeng Zhang, Erjia Xiao, Yuchen Zhang, Haoxiang Fu, Ruibin Hu, Yanbiao Ma, Wenbo Ding, Long Chen, Hangjun Ye, and Xiaoshuai Hao. Team Xiaomi EV-AD VLA: Caption-guided retrieval system for cross-modal drone navigation – technical report for IROS 2025 RoboSense challenge track 4. *arXiv preprint arXiv:2510.02728*, 2025a.

Weichen Zhang, Chen Gao, Shiquan Yu, Ruiying Peng, Baining Zhao, Qian Zhang, Jinqiang Cui, Xinlei Chen, and Yong Li. CityNavAgent: Aerial vision-and-language navigation with hierarchical semantic planning and global memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025b.

Weifan Zhang, Tingguang Li, and Yuzhen Liu. MAG-Nav: Language-driven object navigation leveraging memory-reserved active grounding. *arXiv preprint arXiv:2508.05021*, 2025c.

Guiyong Zheng, Yueting Ban, Mingjie Zhang, Juepeng Zheng, and Boyu Zhou. OnFly: Onboard zero-shot aerial vision-language navigation toward safety and efficiency. *arXiv preprint arXiv:2603.10682*, 2026a.

Wancai Zheng, Hao Chen, Xianlong Lu, Linlin Ou, and Xinyi Yu. 3DGSNav: Enhancing vision-language model reasoning for object navigation via active 3D gaussian splatting. *arXiv preprint arXiv:2602.12159*, 2026b.

A Additional Details for Selective Localization

This appendix provides additional details for the Selective Target Localization Agent. We first explain how counterfactual sparse-view supervision (CSS) constructs partial views and assigns LOCALIZE/DEFER targets. We then describe training, evaluation, and the human-annotation analysis.

A.1 Overview of Counterfactual Sparse-View Supervision

CSS constructs partial observations by masking spatial regions of a world-aligned canvas. Nested views progressively reveal additional visual context while keeping the scene, instruction, and target fixed. A view receives LOCALIZE supervision when the target and selected instruction-relevant cues are observed; otherwise, it receives DEFER supervision with a rationale describing the unobserved target region or context. These annotation-derived criteria are used only to construct offline supervision and are not applied as rules during navigation.

A.2 Sparse-View Construction

The purpose of this construction is to vary the observed context without changing the underlying localization problem. We use 8,000 train_seen episodes, each contributing four partial views of the same world-aligned canvas X. The ground-truth target is used only to build these training views and remains unavailable during navigation.

Complete canvas. We first construct a common canvas so that all partial views of an episode share the same spatial frame. The scene is divided into $100\,\mathrm{m} \times 100\,\mathrm{m}$ cells, and we select a 2×2 window containing the target. When several windows are available, we prefer the one containing more instruction-referenced landmarks and same-type objects. The resulting $200\,\mathrm{m} \times 200\,\mathrm{m}$ canvas is rendered at 1024×1024 pixels and divided into four quadrants.

Sparse-view construction. For the reported experiments, we construct four nested views of each canvas. The initial view reveals a non-target quadrant, and the remaining quadrants are revealed in a fixed cyclic order until the canvas is fully observed. Unobserved regions are filled with the fixed placeholder used during navigation. Supervision is assigned separately to each view. A fully observed canvas can still receive DEFER when a referenced landmark extends beyond its boundary.

A.3 Localization and Deferral Target Construction

The purpose of target construction is to assign consistent supervision as the observed region changes. Following the definition in the main text, a view receives LOCALIZE only when the target and the selected instruction-relevant cues are observed. The cues considered here are annotated reference landmarks and same-type structural context associated with supported relational or ordinal descriptions.

Target and landmark coverage. We first determine whether the target and instruction-referenced landmarks fall within the observed region. The target is treated as observed when its designated cell is visible. For a resolved reference landmark, we measure the fraction of its annotated cells that are observed. A view is assigned DEFER when none of these cells is observed or when fewer than half of them are covered. This fraction is used only for offline supervision; the localization agent does not receive it.

Same-type structural context. We next consider same-type objects because ordinal and structural descriptions may depend on objects outside the current view. Same-type structural context is treated as partial when at least one same-type object is observed and another remains outside the observed region. This condition is used only when the instruction contains a corresponding ordinal, building-row, or parking-related cue. If the required object annotations are unavailable, this condition is not applied.

Language cues. The supported cues are detected from the instruction text. They cover ordinal expressions (e.g., *first*, *last*, or a numbered position), building-row descriptions, and parking-related descriptions. These cue types match the reference landmarks and same-type structural context described in the main text; they are not intended to cover every possible linguistic ambiguity.

Target assignment. The assignment follows the same two outputs used by the localization agent. An unobserved target receives `DEFER`. A visible target also receives `DEFER` when a selected reference landmark or same-type structural context is incomplete under the conditions above. All remaining views receive `LOCALIZE`. Each `DEFER` target is paired with a short rationale describing the corresponding unobserved target region or context. These annotation-derived criteria provide proxy supervision and do not exhaustively determine whether a target is uniquely identifiable.

A.4 Training Targets and Optimization

Output schemas. The training targets follow the same response schemas used during closed-loop navigation. A `LOCALIZE` target contains the target box, whereas a `DEFER` target contains structured deferral information and a natural-language rationale. The rationale directly describes the context associated with the assignment, such as an unobserved target region, a partially observed reference landmark, or incomplete row or parking context. The templates do not include coverage values or other construction metadata.

Bounding-box targets. The localization target should be consistent with the normalized canvas coordinates used by the agent. We therefore project the annotated target contour onto the canvas and convert the resulting box to $[0, 1]^4$. We retain a box only when its center maps within the CityNav success threshold of $20\,\mathrm{m}$ from the annotated target location.

Training-set composition. The 8,000 training episodes yield 32,000 views, with an equal number of views at each stage of the nested construction. The set contains 17,623 target-unobserved `DEFER` views, 6,698 target-visible `DEFER` views, and 7,679 `LOCALIZE` views. The proportion of `DEFER` targets decreases from 100.00% in the initial views to 21.24% in the fully observed canvases, reflecting the correlation between coverage and supervision targets under this construction.

Optimization. We fine-tune Qwen3-VL-2B-Instruct with LoRA and weighted autoregressive cross-entropy. Target-unobserved `DEFER` views receive weight 0.5, and the remaining views receive weight 1.0. The Rationale-Guided Exploration Agent retains the original frozen checkpoint. Additional optimization details are provided in Appendix E.

A.5 Evaluation Protocol

This evaluation assesses the localization agent on constructed partial observations using the supervision criteria described above. We use the same 20,660 sparse views pooled across Validation Seen and Validation Unseen and give every model the same inference prompt. Following the CSS construction, the set contains 10,922 views with an unobserved target, 4,342 views with a visible target but a missing selected instruction-relevant cue, and 5,396 `LOCALIZE` views.

We use the metric definitions introduced in the main experiments. Macro-F1 measures the binary `LOCALIZE`/`DEFER` decision. FLR measures `LOCALIZE` predictions on `DEFER` views, whereas FLR_2 restricts this metric to cases in which the target is visible but a selected instruction-relevant cue is missing. UDR measures `DEFER` predictions on `LOCALIZE` views. Conditional localization accuracy is the percentage of predicted boxes that fall within $20\,\mathrm{m}$ of the target when the model selects `LOCALIZE` on a `LOCALIZE` view. An always-defer model obtains zero FLR but 100% UDR, which is why the two types of error are reported together. The constructed-view results quantify performance under this sampling and labeling protocol. They do not by themselves separate the influence of instruction-specific context from that of observation coverage.

A.6 Human-Annotation Evaluation

Annotation protocol. The human-annotation evaluation examines whether the relative model performance persists when predictions on the same constructed validation views are evaluated against

human judgments rather than the annotation-derived CSS labels. We sample 500 validation views from the three conditions above: 100 with an unobserved target, 200 with a visible target but a missing selected cue, and 200 LOCALIZE views. Annotator A labels all examples using only the instruction and partial observation, without seeing the CSS targets or model predictions. Annotator B independently labels 157 examples, including a 100-example subset sampled as 20/40/40 from the three conditions for agreement analysis. Model results use Annotator A's labels on all 500 examples.

Inter-annotator agreement. We first measure whether the annotation itself is consistent. On the 100-example subset, the annotators agree on 81.0% of the three-way judgments, with Cohen's $\kappa = 0.682$. Table 5 further separates target visibility from the LOCALIZE/DEFER decision. Target visibility has perfect agreement, whereas deciding between LOCALIZE and DEFER for a visible target is less consistent.

Table 5: Inter-annotator agreement on the 100-view subset. The final row is restricted to the 80 views that both annotators regard as target-visible.

Decision boundary	n	Agreement (%)	Cohen's κ
Three-way annotation	100	81.0	0.682
Target visibility	100	100.0	1.000
LOCALIZE/DEFER	100	81.0	0.617
Target-visible LOCALIZE/DEFER	80	76.25	0.453

Sampling-weighted evaluation. The three conditions are sampled at different rates, so we weight the annotated examples to match their frequencies in the full evaluation set. For a view from condition h, the weight is $w_i = N_{h_i}/n_{h_i}$, where N_h and n_h are its counts in the full and annotated sets. Here, $(N_1, N_2, N_3) = (10{,}922, 4{,}342, 5{,}396)$ and $(n_1, n_2, n_3) = (100, 200, 200)$, giving weights of 109.22, 21.71, and 26.98. Macro-F1 is computed from the weighted binary decision-confusion matrix, and the same weights are used for the remaining metrics. All models use the same human labels, weights, and previously generated predictions.

Results. The purpose of this comparison is to check whether the model ranking remains under human judgments. Table 6 shows the same overall pattern as the CSS-based evaluation. Our method improves Macro-F1 and lowers both FLR measures relative to bbox-only SFT, while maintaining comparable conditional localization accuracy. Its higher UDR indicates the same trade-off observed in the main paper: the agent predicts fewer boxes on DEFER views but also defers more often on LOCALIZE views.

Table 6: Selective localization evaluated against human annotations on 500 validation views. Sampling weights account for the different sampling rates of the three conditions. Definitions match Table 2; bold indicates the best value among the three models.

Model	Macro-F1 (%)↑	FLR (%)↓	FLR_2 (%)↓	Cond. Acc. (%)↑	UDR (%)↓
Base VLM (2B)	55.09	13.04	22.93	22.45	75.72
Bbox-only SFT	55.13	40.33	35.40	43.13	**46.60**
Ours	**64.67**	**5.16**	**10.73**	**45.72**	67.34

B Sensitivity to the Exploration Budget

This analysis examines how much additional observation is useful before further exploration yields diminishing returns. We vary T from 0 to 5 on the original CityNav Test Unseen split while keeping all other settings unchanged. Here, T counts exploratory movements after landmark-based initialization. A predicted box triggers terminal navigation; if the agent still outputs DEFER after T movements, the UAV stops at its current position. Success uses the standard $20\,\mathrm{m}$ threshold, and path length includes initialization, exploration, and terminal navigation.

Table 7 shows that most of the improvement occurs within the first two exploratory movements. Larger budgets provide smaller additional gains in SR and SPL, while the average number of executed moves remains low because exploration stops after localization. NE does not improve with a larger budget, indicating that additional movements mainly convert some failures into successes rather than uniformly reducing final error.

Table 7: Sensitivity to the exploration budget on the original CityNav Test Unseen split. SR and SPL are percentages; NE is in meters. Avg. steps is the mean number of executed exploratory movements over all episodes, including failures and episodes with zero exploration.

Budget T	SR↑	SPL↑	NE↓	Avg. steps
0	34.48	25.62	34.53	0.00
1	39.58	29.11	34.67	0.70
2	41.26	30.21	35.51	0.98
3	41.90	30.55	35.53	1.12
4	42.45	30.68	35.65	1.20
5	42.57	30.95	35.74	1.22

C Qualitative Analysis

We use qualitative examples to illustrate both the closed-loop interaction and the remaining failure modes. Figure 4 provides an additional successful interaction. The localization agent first defers because the street context is incomplete. The exploration agent uses this rationale and the landmark priors to inspect Northampton Street; after the observation is added to memory, the localization agent selects the target car beside Westminster College.

The framework can still fail after exploration, typically when visually similar candidates remain or when spatial and ordinal relations are misinterpreted. Figure 5 shows two such cases; both move from `DEFER` to an incorrect `LOCALIZE` prediction.

Incorrect instance under compound spatial constraints. In Fig. 5(a), the instruction identifies a red car by both its ordinal position within a row and the location of the parking lot behind the backyards of houses on Livingstone Road. After a `LEFT` action, the agent selects a red car outside the specified parking lot. Although more of the row becomes visible, the model does not correctly combine the parking-lot relation with the ordinal cue.

Incorrect landmark-relative region. In Fig. 5(b), the instruction refers to the rectangular parking lot in front of Perry Barr Bus Depot. After observing more of the area, the agent predicts a parking region on the wrong side of the depot, indicating an incorrect interpretation of *in front of*.

Implications. Additional observation does not ensure that every instruction constraint is interpreted correctly. The current framework reassesses localization from the updated memory but does not separately verify that the rationale has been resolved before accepting a box. This can end exploration at an incorrect candidate.

D Successful Navigation Trajectories

Fig. 6 complements the agent interactions in Fig. 3 with three successful navigation trajectories on CityNav. Each executed trajectory includes landmark-based initialization, exploration, and terminal navigation to the predicted target location. In these episodes, the UAV terminates within the 20 m success threshold of the corresponding ground-truth target.

E Implementation Details

This section reports the training and inference settings needed to reproduce the framework while following the notation and agent inputs defined in the main method.

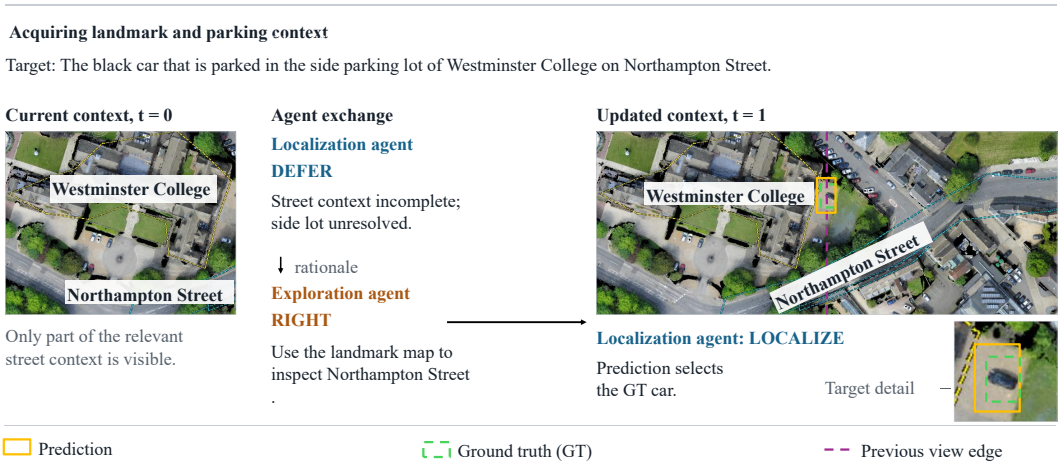

Figure 4: Additional successful interaction on CityNav. The localization agent defers because the relevant street context is incomplete; the exploration agent then uses this rationale and the landmark priors to select the next movement. After the new observation is added to memory, the localization agent predicts the target car beside Westminster College. Colors and line styles follow Fig. 3. Search actions are relative to the UAV heading.

Training. We use a lightweight LoRA fine-tuning setup for the Selective Target Localization Agent. The model is trained for one epoch with AdamW, cosine learning-rate decay, a learning rate of 2×10^{-5}, and a warm-up ratio of 0.03. LoRA uses rank 16, $\alpha = 32$, and dropout 0.05. Training uses three NVIDIA RTX 3090 GPUs, an effective batch size of 6, `bfloat16`, gradient checkpointing, and random seed 7.

Online evaluation. Online evaluation uses the same metric scale for local observations, spatial memory, and search movements. Each observation covers $100\,\mathrm{m} \times 100\,\mathrm{m}$ at 512^2 pixels, and the retained canvas covers $200\,\mathrm{m} \times 200\,\mathrm{m}$ at 1024^2 pixels. Search is initialized from the instruction-referenced landmark priors when available and otherwise from the initial observation. The reported experiments use an exploration step size of $100\,\mathrm{m}$ and a default budget of $T = 5$ movements. Appendix B varies the movement budget while keeping the step size fixed. Both agents use greedy decoding with a maximum of 256 generated tokens.

Prompts and outputs. The prompts follow the inputs defined in the main method. The localization agent receives the instruction and world-aligned visual context and returns either a normalized target box or a deferral response with rationale r_t. After a deferral, the exploration agent receives the instruction, rationale, shared spatial context, UAV pose, and target-blind landmark priors. It selects one heading-relative action from $\{\texttt{forward}, \texttt{backward}, \texttt{left}, \texttt{right}\}$.

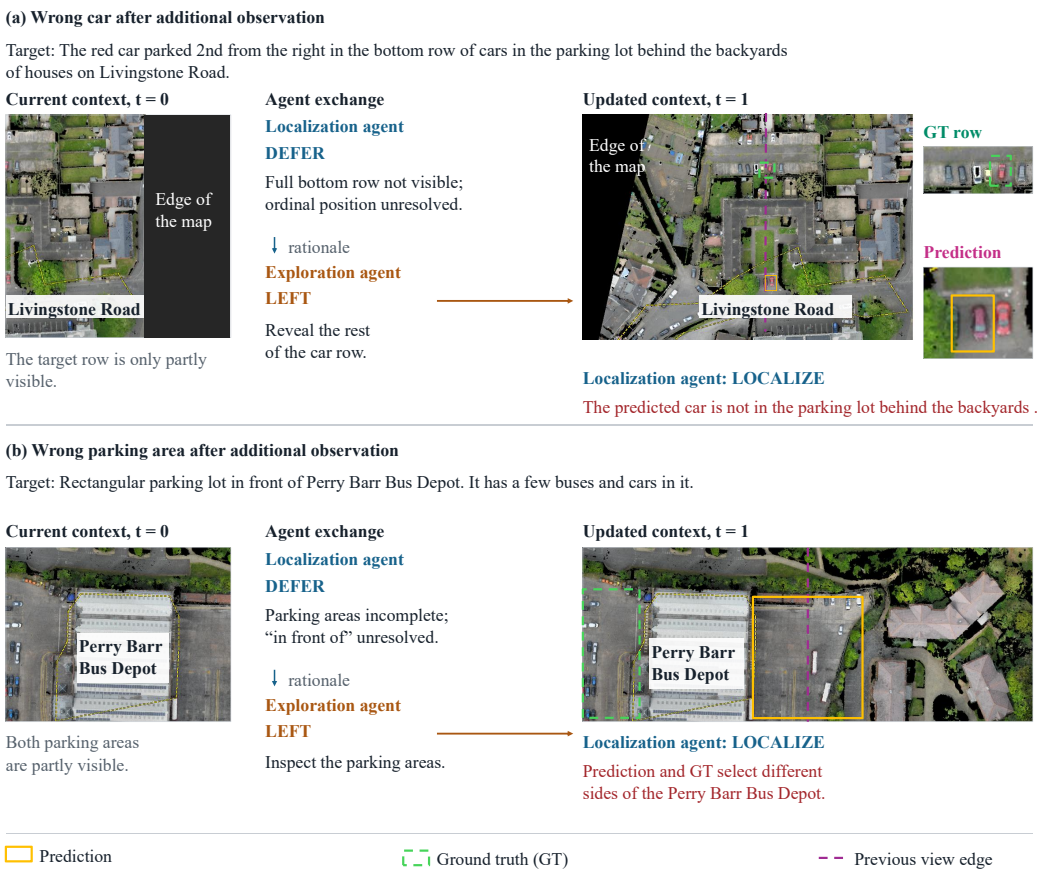

Figure 5: Failure cases after exploration. In both examples, the localization agent first defers and then predicts an incorrect box from the updated memory. (a) The selected car does not satisfy both the parking-lot and ordinal relations. (b) The selected parking area is on the wrong side of the depot. Colors and line styles follow Fig. 3. Search actions are relative to the UAV heading.

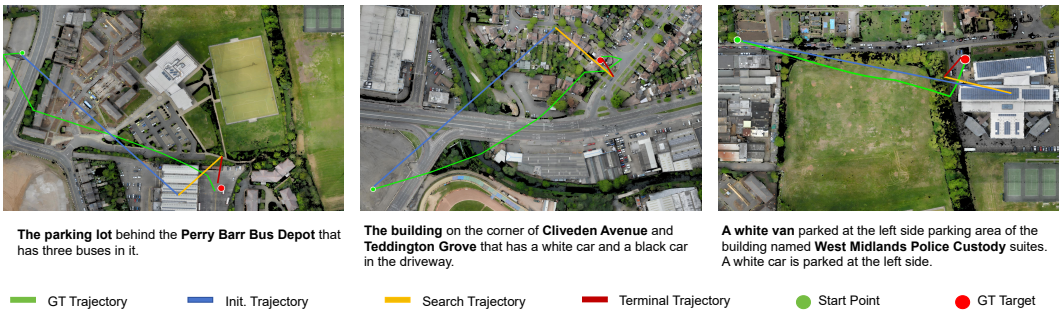

Figure 6: Successful navigation trajectories on CityNav. From left to right, the targets are a parking lot behind Perry Barr Bus Depot, a building at the corner of Cliveden Avenue and Teddington Grove, and a white van beside West Midlands Police Custody suites. Green lines show ground-truth reference trajectories. Blue, yellow, and red lines show the executed initialization, search, and terminal navigation segments, respectively. Green and red circles mark the starting positions and ground-truth targets, respectively.